AMENDMENTS TO THE CLAIMS

Kindly amend claims 1, 6, and 11 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) In a processor having multiple hardware streams supporting multiple data threads, and a data cache, a system for fetching instructions from one to P of the multiple hardware streams to a pipeline, where P is less than the number of multiple hardware streams, the system comprising:

multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast whether corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache, wherein said multiple hit/miss predictors forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache prior to when said corresponding instructions enter into a dispatch stage in the ~~pipeline; and~~pipeline;

a fetch stage, coupled to said multiple hit/miss predictors, configured to simultaneously fetch every cycle, the instructions from the one to P of the multiple hardware streams to the pipeline, and configured to select, on a cycle-by-cycle basis, the one to P of the multiple hardware streams from which to fetch the instructions; and

an instruction scheduler, coupled to said fetch stage, for managing access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, wherein at any point in time, said instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads.

2. (Previously Presented) The system as recited in claim 1, wherein a hit prediction precipitates no change in the fetching of the instructions.

3. (Previously Presented) The system as recited in claim 1, wherein a miss prediction results in switching the fetching to different ones of the multiple hardware streams.

4. (Previously Presented) The system as recited in claim 1, wherein said each of said multiple hit/miss predictors generates a confidence level value, and said confidence level value is used by said fetch algorithm to select the P of the multiple hardware streams.

5. (Previously Presented) The system as recited in claim 1, wherein said multiple hit/miss predictors further operate at a dispatch level to optimize the dispatch of consumer instructions by predicting latency of data.

6. (Currently Amended) A processor having multiple hardware streams supporting multiple data threads, the processor comprising:

a data cache, comprising a plurality of levels;

multiple hit/miss predictors, each associated with a corresponding one of the multiple hardware streams, said each configured to forecast whether corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said data cache, , wherein said multiple hit/miss predictors forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said data cache prior to when said corresponding instructions enter into a dispatch stage in a pipeline of the processor, said each of said multiple hit/miss predictors comprising:

a plurality of hit/miss predictors, each configured to forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss one or more of said levels;

a fetch stage, coupled to said multiple hit/miss predictors, for simultaneously fetching every cycle, instructions from one to P of the multiple hardware streams, wherein P is less than the number of the multiple hardware streams, and configured to select, on a cycle-by-cycle basis, said one to P of the multiple hardware streams from which to fetch said instructions, wherein said ~~fetch algorithm~~fetch stage selects said one to P of the multiple hardware streams based upon whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said one or more of said levels; and

an instruction scheduler, coupled to said fetch stage, for managing access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, wherein at any point in time, said instruction scheduler manages access for a given one of the multiple hardware streams according to a priority record, regardless of any priority associated with the multiple data threads.

7. (Previously Presented) The processor as recited in claim 6, wherein a hit prediction precipitates no change in the fetching of said instructions.

8. (Previously Presented) The processor as recited in claim 6, wherein a miss prediction results in switching the fetching to different ones of the multiple hardware streams.

9. (Previously Presented) The processor as recited in claim 6, wherein said each of said multiple hit/miss predictors generates a confidence level value, and said confidence level value is used by said fetch stage to select said one to P of the multiple hardware streams.

10. (Previously Presented) The processor as recited in claim 6, wherein said multiple hit/miss predictors further operate at a dispatch level to optimize the dispatch of consumer instructions by predicting latency of data.

11. (Currently Amended) In a processor having multiple hardware streams supporting multiple data threads, and a data cache, a method for simultaneously fetching instructions every cycle from one to P of the multiple hardware streams to a pipeline, where P is less than the number of the multiple hardware streams, the method comprising:

for each of the multiple hardware streams, making a hit/miss prediction by a corresponding one of associated hit/miss predictors as to whether corresponding instructions for the each of the multiple hardware streams previously fetched will hit or miss the data cache, , wherein said making is performed prior to when the corresponding instructions enter into a dispatch stage in the pipeline; and

selecting, on a cycle-by-cycle basis, the one to P of the multiple hardware streams from which to fetch the instructions; and

managing access for the multiple hardware streams to a set of functional resources for processing instructions from the multiple hardware streams, wherein at any point in time, said managing for a given one of the multiple hardware streams is accomplished according to a priority record, regardless of any priority associated with the multiple data threads.

12. (Previously Presented) The method as recited in claim 11, wherein said making comprises:

generating a confidence level value, and employing the confidence level to select the one to P of the multiple hardware streams.

13. (Previously Presented) The method as recited in claim 11, further comprising:

further operating the multiple hit/miss predictors at a dispatch level to optimize the dispatch of consumer instructions by predicting latency of data.

14. (Previously Presented) The system as recited in claim 1, wherein the processor comprises a fine-grained multistreaming processor that concurrently executes the instructions from the multiple hardware streams.

15. (Previously Presented) The system as recited in claim 1, wherein the data cache comprises:

a first level and a second level, and wherein said each of said multiple hit/miss predictors comprises:

a first hit/miss predictor, configured to forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said first level; and

a second hit/miss predictor, configured to forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said second level;

wherein said fetch stage selects the one to P of the multiple hardware streams based upon whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss said second level.

16. (Previously Presented) The system as recited in claim 1, wherein the processor comprises a network processor, and wherein said each of said multiple hit/miss predictors employs a flow number to which a packet belongs to forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache.

17. (Previously Presented) The processor as recited in claim 6, wherein the processor comprises a network processor, and wherein said each of said multiple hit/miss predictors employs a flow number to which a packet belongs to forecast whether said corresponding instructions from said corresponding one of the multiple hardware streams will hit or miss the data cache.

18. (Previously Presented) The method as recited in claim 11, wherein said selecting comprises:

switching the fetching to a different one to P of the multiple hardware streams.

19. (Previously Presented) The method as recited in claim 11, wherein the data cache comprises a first level and a second level, and wherein said making comprises:

first forecasting whether said corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the first level; and

second forecasting whether the corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the second level; and

wherein said selecting comprises:

choosing the one to P of the multiple hardware streams based upon whether the corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the second level.

20. (Previously Presented) The method as recited in claim 11, wherein said making comprises:

employing a flow number to which a packet belongs to forecast whether the corresponding instructions from the corresponding one of the multiple hardware streams will hit or miss the data cache.